Gilbert bound
y, vol. IT-33,

put statistics,"
ay 1993.
:ntification via
25, Jan. 1992.
ular curves and
form. Theory,

umber genera-
cent Results in
s, 1976.
rsity, load bal-
48, Apr. 1989.
mat packings,"

Decoding Algebraic–Geometric Codes up to the Designed Minimum Distance

Gui-Liang Feng and T. R. N. Rao, *Fellow, IEEE*

Abstract— **A simple decoding procedure for algebraic–geometric codes $C_\Omega(D, G)$ is presented. This decoding procedure is a generalization of Peterson's decoding procedure for the BCH codes. It can be used to correct any $\lfloor (d^* - 1)/2 \rfloor$ or fewer errors with complexity $O(n^3)$, where d^* is the designed minimum distance of the algebraic–geometric code and n is the codelength.**

Index Terms— **Error-correcting codes, algebraic–geometric codes, decoding procedure, correcting $\lfloor (d^* - 1)/2 \rfloor$ errors.**

I. Introduction

THE MOST important development in the theory of error-correcting codes in recent years is the introduction of methods from algebraic geometry to construct linear codes. These so called *algebraic–geometric codes* were introduced by Goppa. In 1982, Tsfasman, Vlăduţ and Zink [1] obtained an extremely exciting result: the existence of a sequence of codes that exceeds the Gilbert–Varshamov bound [2]. For this paper, they received the IEEE Information Theory Group Paper Award for 1983. Since then, many papers dealing with algebraic–geometric codes have followed [3]–[10].

Good code constructions are very important. Moreover, it is desirable and important to derive simple decoding procedures which can correct as many errors as possible. Justesen *et al.* [11] first presented a decoding procedure for codes from nonsingular plane algebraic curves. This decoding procedure can only correct $\lfloor (d^* - g - 1)/2 \rfloor$ or fewer errors, where d^* is the designed minimum distance of the code and g is the genus of the curve involved in the construction. Skorobogatov and Vlăduţ [12] generalized their ideas and gave a decoding procedure which can correct any $\lfloor (d^* - g - 1)/2 \rfloor$ or fewer errors for codes from arbitrary algebraic curves. In their paper, Skorobogatov and Vlăduţ also presented a modified algorithm, correcting more errors, but in general, not up to the designed minimum distance. Using profound results from algebraic geometry, Pellikaan [13] gave a decoding procedure which decodes up to $\lfloor (d^* - 1)/2 \rfloor$ errors. However, his decoding procedure is very complex and is not completely effective. Recently, Justesen *et al.* [14] improved on their original decoding procedure in several ways and gave a new decoding procedure for codes from arbitrary regular plane curves, which can decode up to $\lfloor (d^* - g/2 - 1)/2 \rfloor$ errors.

Manuscript received November 5, 1991; revised May 20, 1992. This work was supported in part by the Office of Naval Research under Grant N00014-91-J-1067. This work was presented in part at the 9th International Symposium on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes, New Orleans, LA, October 1991.

The authors are with the Center for Advanced Computer Studies, University of Southwestern Louisiana, Lafayette, LA 70504.

IEEE Log Number 9203437.

In this paper, we present a fairly simple decoding procedure capable of decoding up to $\lfloor (d^* - 1)/2 \rfloor$ errors. The improvement is obtained by using a form of majority scheme to find unknown syndromes in the well-known algorithm. The procedure can be implemented easily by hardware or software.

The paper is organized as follows. In the next section, for easy reference, we include a fundamental iterative algorithm (FIA), which is very similar to the Gaussian elimination and can be used to easily derive the Berlekamp–Massey algorithm and the generalized Berlekamp–Massey algorithm [16]. Then we modify the FIA and give some related properties, which will be used in other sections. In Section III, a new decoding procedure for algebraic–geometric codes $C_\Omega(D, G)$ with $G = mQ$ is presented. In order to easily understand this decoding procedure, one example is shown in Section IV. Finally, some conclusions are given in Section V.

II. Fundamental Iterative Algorithm

In this section, the fundamental iterative algorithm (FIA) [16] is modified. This modified algorithm is our main algorithm for decoding algebraic–geometric codes up to the designed minimum distance. To a certain extent it is similar to the Berlekamp–Massey algorithm, which is the main algorithm for decoding BCH codes up to the designed minimum distance. For easy reference, the FIA is described briefly in the following. This algorithm is for finding the smallest initial set of dependent columns in a matrix over any field F. That is, let

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ | & | & \cdots & | \\ a_{M1} & a_{M_2} & \cdots & a_{MN} \end{bmatrix}$$

be such a matrix, to find the smallest l and $c_1, \cdots, c_l$ such that

$$a_{i,l+1} + c_1 \cdot a_{i,l} + \cdots + c_l \cdot a_{i,1} = 0 \qquad \text{for } i = 1, 2, \cdots, M. \tag{2.1}$$

For each column j, let $C^{(i-1,j)}(x) = \sum_{k=0}^{j-1} c_k^{(i-1,j)} x^k$, where $c_0^{(i-1,j)} = 1$, be defined as the polynomial with the property that

$$\left[C^{(i-1,j)}(x) \cdot a^{(h)}(x)\right]_j = a_{h,j} + c_1^{(i-1,j)} a_{h,j-1} + \cdots + c_{j-1}^{(i-1,j)} a_{h,1} = 0 \qquad \text{for } h \leqslant i - 1, \tag{2.2}$$

where $a^{(h)}(x) = 1 + a_{h,1}x + a_{h,2}x^2 + \cdots + a_{h,N}x^N$ and $[f(x)]_j$ expresses the coefficient of x^j in $f(x)$. Accordingly, the initial polynomial for column j is designated $C^{(0,j)}(x)$ with $C^{(0,1)}(x) = 1$ as the initial polynomial for the first column. Let

$$d_{i,j} = \left[C^{(i-1,j)}(x) \cdot a^{(i)}(x)\right]_j = a_{i,j} + c_1^{(i-1,j)}a_{i,j-1} + \cdots + c_{j-1}^{(i-1,j)}a_{i,1} \qquad (2.3)$$

be the discrepancy at row i and column j. For some column j, if $d_{i,j} = 0$ for $i = 1, 2, \cdots, r-1$, then we have $C^{(0,j)}(x) = C^{(1,j)}(x) = \cdots = C^{(r-1,j)}(x)$. Suppose $d_{r,j} \neq 0$ and there is no u, where $1 \leqslant u < j$ and $C^{(u)}(x) = C^{(r-1,u)}(x)$, then we define the final polynomial of column j, $C^{(j)}(x) = C^{(r-1,j)}(x)$, at the location of $a_{r,j}$ and place a cross "×" to indicate that the discrepancy is nonzero (we call it the final discrepancy of column j; the discrepancies at the locations of $a_{i,j}$ for $i < r$ are all zero), and then move on the next column. We then start examining the top element of the next column with $C^{(0,j+1)}(x) = C^{(j)}(x) = C^{(r-1,j)}(x)$. Otherwise, there is a $C^{(u)}(x)$ at a column u, where $1 \leqslant u < j$ such that $C^{(u)}(x) = C^{(r-1,u)}(x)$. Then $C^{(r,j)}(x)$ can be obtained by the following equation

$$C^{(r,j)}(x) = C^{(r-1,j)}(x) - \frac{d_{r,j}}{d_{r,u}} C^{(u)}(x)x^{j-u}. \qquad (2.4)$$

The following can be easily seen.

Lemma 1: There is an "×" in column j, if and only if it is linearly independent on its previous columns.

Thus, beginning with the first column, we examine the elements in successive columns of matrix A, one by one, from the top row towards the bottom row. If the problem has a solution for A, $C^{(M,l)}(x)$ is a solution. From [16], it is easily seen that if $d_{i,j} = 0$, then the subvector of the top i components of column j is a linear combination of the subvectors of the top i components of its previous $j-1$ columns. Hereinafter, it is said that column j is a *partial linear combination of its previous columns with the top i components.* The linear combination coefficients are the coefficients of $C^{(i,j)}(x)$.

Thus, we have the following iterative algorithm, which is called the fundamental iterative algorithm (FIA). Two storage tables, D and C, are set up for this algorithm. Table D stores the discrepancies $d_{r,j}$, the final discrepancy of column j, while Table C stores the corresponding polynomials $C^{(j)}(x)$, the final polynomial of column j.

Fundamental Iterative Algorithm:

Step 1) Empty Tables D and C, $1 \rightarrow j$, $1 \rightarrow r$, $1 \rightarrow C^{(j)}(x)$.

Step 2) Compute $d_{r,j} = \left[C^{(j)}(x) \cdot a^{(r)}(x)\right]_j$.

Step 3) IF $d_{r,j} = 0$, THEN

a) IF $r = M$, then $j - 1 \rightarrow l$, $C^{(j)}(x) \rightarrow C(x)$, STOP; we have a solution.

b) otherwise $r + 1 \rightarrow r$, and return to Step 2).

Step 4) IF $d_{r,j} \neq 0$, THEN

a) IF there exists a $d_{r,u} \in D$ for some $1 \leqslant u < j$, THEN

$$C^{(j)}(x) - \frac{d_{r,j}}{d_{r,u}} C^{(u)}(x)x^{j-u} \rightarrow C^{(j)}(x)$$

and return to Step 3).

b) otherwise, $d_{r,j}$ is stored in D, $C^{(j)}(x) \rightarrow C^{(j+1)}(x)$ and $C^{(j)}(x)$ is stored in C, mark "×" at row r and column j. IF $j < N$, THEN $j + 1 \rightarrow j$, $1 \rightarrow r$, and return to Step 2). Otherwise STOP, as this problem has no solution.

Now let us consider another related problem. Let S' be such a matrix: (see matrix at bottom of page) where @ and # are all unknown, and other elements are known. In each column j, if $s_{i,j}$ is @, then the values of $s_{u,j}$ are known for $u < i$ and $s_{v,j}$ are all # for $v > i$; if there is no @ and $s_{i,j}$ is the first #, then the elements above $s_{i,j}$ are known and the elements below $s_{i,j}$ are all #. Similarly, for each row i, if $s_{i,j}$ is an @, then the value of $s_{i,u}$ are all known for $u < j$, and $s_{i,v}$ are all # for $v > j$; if there is no @ and $s_{i,j}$ is the first #, then the left elements of $s_{i,j}$ are known and the right elements are all #.

Suppose column j is partial linear combination of its previous columns with the top $i-1$ components and $s_{i,j}$ is an @, we want to know whether there is a unique value of @ at (i,j) such that column j is a partial linear combination of its previous columns with the top i components. If there is, how is this unique value determined? The key step in decoding some linear error correcting codes can be reduced to this problem

$$\begin{bmatrix}
s_{1,1} & s_{1,2} & s_{1,3} & s_{1,4} & s_{1,5} & s_{1,6} & s_{1,7} & s_{1,8} & s_{1,9} & s_{1,10} & s_{1,11} \\
s_{2,1} & s_{2,2} & s_{2,3} & s_{2,4} & s_{2,5} & s_{2,6} & s_{2,7} & s_{2,8} & s_{2,9} & @ & \# \\
s_{3,1} & s_{3,2} & s_{3,3} & s_{3,4} & s_{3,5} & s_{3,6} & s_{3,7} & s_{3,8} & @ & \# & \# \\
s_{4,1} & s_{4,2} & s_{4,3} & s_{4,4} & s_{4,5} & s_{4,6} & @ & \# & \# & \# & \# \\
s_{5,1} & s_{5,2} & s_{5,3} & s_{5,4} & s_{5,5} & s_{5,6} & \# & \# & \# & \# & \# \\
s_{6,1} & s_{6,2} & s_{6,3} & s_{6,4} & s_{6,5} & @ & \# & \# & \# & \# & \# \\
s_{7,1} & s_{7,2} & s_{7,3} & s_{7,4} & @ & \# & \# & \# & \# & \# & \# \\
s_{8,1} & s_{8,2} & @ & \# & \# & \# & \# & \# & \# & \# & \# \\
s_{9,1} & @ & \# & \# & \# & \# & \# & \# & \# & \# & \# \\
\# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \#
\end{bmatrix}$$

(See next section): finding a partial linear combination with the top R components for a given integer R (if there is) and find all @, which are uniquely determined by the above requirement. In order to solve this problem, we modify the FIA as follows:

2') The algorithm stops when $C^{(R,j)}(x)$ or $C^{(N)}(x)$ is obtained.

2'') Once $C^{(i-1,j)}(x)$ is obtained, where $s_{i,j}$ is an @ or an #, then the algorithm defines the final polynomial of column j, $C^{(j)}(x) = C^{(i-1,j)}(x)$, and moves to the top element of the next column, namely, $s_{1,j+1}$.

Thus, after applying the modified FIA, the matrix $D = (d_{ij})$, called the discrepancy matrix, is obtained. The discrepancy matrix looks like the following:

$$\begin{bmatrix} \times & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s_{2,1} & 0 & 0 & \times & 0 & 0 & 0 & 0 & @ & \# \\ s_{3,1} & 0 & \times & s_{3,4} & 0 & 0 & 0 & 0 & @ & \# \\ s_{4,1} & 0 & s_{4,3} & s_{4,4} & 0 & 0 & @ & \# & \# & \# \\ s_{5,1} & \times & s_{5,3} & s_{5,4} & 0 & 0 & \# & \# & \# & \# \\ s_{6,1} & s_{6,2} & s_{6,3} & s_{6,4} & \times & @ & \# & \# & \# & \# \\ s_{7,1} & s_{7,2} & s_{7,3} & s_{7,4} & @ & \# & \# & \# & \# & \# \\ s_{8,1} & s_{8,2} & @ & \# & \# & \# & \# & \# & \# & \# \\ s_{9,1} & @ & \# & \# & \# & \# & \# & \# & \# & \# \\ \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \end{bmatrix}.$$

From this matrix, it is easily to see the following.

2a) If there is no "×" in column j, then column j is a partial linear combination of its previous columns with the top $i-1$ components, where i is the smallest integer that an @ or # is at (i, j). For the above matrix, $i = 6$, $j = 6$; $i = 4$, $j = 7$; and so on.

2b) Suppose $s_{i,j}$ is an @. This @ can be uniquely determined such that column j is a partial linear combination of its previous columns with the top i components, if and only if there is no "×" at the locations of $s_{i,u}$ for $1 \leqslant u < j$. For example, @ at (4, 7) can be uniquely determined.

2c) If the value of @ at (i, j) is uniquely determined by (2b), then we have

$$c_0^{(i-1,j)} \cdot @ + \sum_{h=1}^{j-1} c_h^{(i-1,j)} \cdot s_{i,j-h} = 0,$$

where

$$C^{(i-1,j)}(x) = \sum_{h=0}^{j-1} c_h^{(i-1,j)} x^h \quad \text{and} \quad c_0^{(i-1,j)} = 1,$$

that is,

$$@ = -\sum_{h=1}^{j-1} c_h^{(i-1,j)} \cdot s_{i,j-h}.$$

Now we formally modify the FIA to solve this problem. We need two more tables, E and F, which store these uniquely determined values of @ and their corresponding $C^{(i,j)}(x)$, respectively. Let $s^{(r)}(x) = 1 + s_{r,1}x + s_{r,2}x^2 + \cdots + s_{r,N}x^N$.

A Modified Fundamental Iterative Algorithm (MFIA):

Step 1) Empty Tables D, C, E, and F, $1 \rightarrow j$, r, and $1 \rightarrow C^{(j)}(x)$.

Step 2) Compute $d_{r,j} = [C^{(j)}(x)s^{(r)}(x)]_j$.

Step 3) IF $d_{r,j} = 0$, THEN

a) IF $r = R$ OR IF $j = N$ and $s_{r+1,N}$ is an @ or #, THEN STOP (according to (2')');

b) IF $s_{r+1,j}$ is an @, $r < R$, and $j < N$, check if there is not $s_{r+1,u} \in D$ for $1 \leqslant u < j$; when it is true, calculate $-\sum_{h=1}^{j-1} c_h^{(j)} \cdot s_{r+1,j-h}$, this value and $C^{(j)}(x)$ are stored in E and F, respectively; then $C^{(j)}(x) \rightarrow C^{(j-1)}(x)$, $j + 1 \rightarrow j$, $1 \rightarrow r$, and return to Step 2) (according to (2'') and 2a)–2c));

c) IF $s_{r+1,j}$ is an #, $r < R$ and $j < N$, then $C^{(j)}(x) \rightarrow C^{(j+1)}(x)$, $j + 1 \rightarrow j$, $1 \rightarrow r$, and return to Step 2) (according to 2');

d) otherwise, $r + 1 \rightarrow r$ and return to Step 2).

Step 4) IF $d_{r,u} \neq 0$, THEN

a) IF there exists a $d_{r,u} \in D$ for some $1 \leqslant u < j$, THEN

$$C^{(j)}(x) - \frac{d_{r,j}}{d_{r,u}} C^{(u)}(x)x^{j-u} \rightarrow C^{(j)}(x)$$

and return to Step 3);

b) otherwise, $d_{r,j}$ is stored in D, $C^{(j)}(x)$ is stored in C, mark "×" at the row r and the column j, $C^{(j)}(x) \rightarrow C^{(j+1)}(x)$, $j + 1 \rightarrow j$, $1 \rightarrow r$ and return to Step 2).

When the algorithm stops, a partial linear combination of column j with the top R components can be found or all values of @, which are uniquely determined by partial linear combinations, can be found. Obviously, the complexity of this algorithm is $O(MN^2)$.

III. A New Decoding Procedure

In this section, we consider the general algebraic–geometric codes $C_\Omega(D, G)$ from X, where X is a nonsingular projective curve over F_q, D is the divisor $P_1 + \cdots + P_n$, $G = m \cdot Q$, and $Q \neq P_i$ for $1 \leqslant i \leqslant n$. The linear code $C_\Omega(D, G)$ of length n over F_q is the image of the linear map $\alpha^*: \Omega(G-D) \rightarrow F_q^n$ defined by

$$\alpha^*(\omega) := (\mathrm{Res}_{P_1}(\omega), \mathrm{Res}_{P_2}(\omega), \cdots, \mathrm{Res}_{P_n}(\omega)).$$

From [15], the parameters of the code are

$$d^* = m - 2g + 2, \qquad k = m - g + 1,$$
$$t = \lfloor (d^* - 1)/2 \rfloor = \lfloor (m+1)/2 \rfloor - g.$$

We may assume that $t > 0$ or $m > 2g$. Recall that a number o_i is nongap for Q if $L(o_iQ) \neq L((o_i - 1)Q)$. Then a function ϕ_{o_i} exists with pole order o_i at Q. As is well known, the nongaps satisfy the following.

Lemma 2: $o_0 = 0$,

$$0 < o_1 < o_2 < \cdots < o_{g-1} < 2g,$$
$$o_i = i + g, \quad \text{for } i = g, g+1, \cdots, m-g.$$

As a basis $f_1, f_2, \cdots, f_{m-g+1}$ of the space $L(G)$, we choose functions $f_i = \phi_{o_{i-1}}$, for $i = 1, 2, \cdots, m+1-g$.

Let $\mathbf{u}$, $\mathbf{e}$, and $\mathbf{c}$ be a received word, an error vector, and a codeword, respectively. Then, we define the syndromes as

$$s_i := (\mathbf{u}, \alpha^* \phi_i) = \sum_{j=1}^{n} u_j \phi_i(P_j) = \sum_{\mu=1}^{v} e_{k_\mu} \phi_i(P_{k_\mu}). \quad (3.1)$$

where v is the number of errors.

Now we define two-dimensional syndromes as

$$S_{i,j} = \sum_{\mu=1}^{v} e_{k_\mu} \phi_{o_{i-1}} \phi_{o_{j-1}}(P_{k_\mu}). \quad (3.2)$$

It is known that if $o_{i-1} + o_{j-1} = o_p \leqslant m$, then $\phi_{o_{i-1}} \cdot \phi_{o_{j-1}} \in L(o_p Q) \subseteq L(mQ)$. Thus, $S_{i,j}$ is a linear combination of $s_0, s_{o_1}, \cdots, s_{o_p}$ and the coefficient of s_{o_p} is not zero. Therefore, all $S_{i,j}$ are known for $o_{i-1} + o_{j-1} \leqslant m$.

On the other hand, $S_{i,j}$ may not be equal to $s_{o_{i-1}+o_{j-1}}$, because $\phi_{o_{i-1}} \phi_{o_{j-1}}$ may not be equal to $\phi_{o_{i-1}+o_{j-1}}$. However, there is a linear relation between them, that is, $S_{i,j}$ can be uniquely determined by s_k for $0 \leqslant k \leqslant o_{i-1} + o_{j-1}$. In this case, we say that $s_{o_{i-1}+o_{j-1}}$ and $S_{i,j}$ are *consistent* and $S_{i,j}$ is a *consistent term* of $s_{o_{i-1}+o_{j-1}}$. As a notation, we use $s'_{o_{i-1}+o_{j-1}}$ to express all consistent terms of $s_{o_{i-1}+o_{j-1}}$. One example will be shown in Section IV. Hereinafter, "the value of $s_{o_{i-1}+o_{j-1}}$" means "the value of $s_{o_{i-1}+o_{j-1}}$ and the values of its possible consistent terms"; "the number of $s_{o_{i-1}+o_{j-1}}$ in matrix S^{**} means "the number of it and its possible consistent terms in S^{**}"; "to substitute @ for $s_{o_{i-1}+o_{j-1}}$" means "to substitute @ for it and its possible consistent terms"; "the number of $s_{o_{i-1}+o_{j-1}}$ with the same value" means "the number of it and its possible consistent terms, whose values satisfy some liner relations"; and so on.

We construct a matrix, $S = (S_{i,j})_{(m-g+1)\times(m-g+1)}$, from Lemma 2:

$$\begin{bmatrix} S_{1,1} & S_{1,2} & S_{1,3} & \cdots & S_{1,m-g} & S_{1,m-g+1} \\ S_{2,1} & S_{2,2} & S_{2,3} & \cdots & S_{2,m-g} & S_{2,m-g+1} \\ | & | & | & & | & | \\ S_{m-g,1} & S_{m-g,2} & S_{m-g,3} & \cdots & S_{m-g,m-g} & S_{m-g,m-g+1} \\ S_{m-g+1,1} & S_{m-g+1,2} & S_{m-g+1,3} & \cdots & S_{m-g+1,m-g} & S_{m-g+1,m-g+1} \end{bmatrix}$$

From (3.1) and (3.2), $S_{1,j} = s_{o_{j-1}}$ and $S_{i,1} = s_{o_{i-1}}$, and S can be decomposed into a product of three matrices as follows:

$$S = X^T \cdot Y \cdot X,$$

where

$$X = \begin{bmatrix} 1 & \phi_{o_1}(P_{k_1}) & \phi_{o_2}(P_{k_1}) & \cdots & \phi_{o_{m-g-1}}(P_{k_1}) & \phi_{o_{m-g}}(P_{k_1}) \\ 1 & \phi_{o_1}(P_{k_2}) & \phi_{o_2}(P_{k_2}) & \cdots & \phi_{o_{m-g-1}}(P_{k_2}) & \phi_{o_{m-g}}(P_{k_2}) \\ | & | & | & & | & | \\ 1 & \phi_{o_1}(P_{k_v}) & \phi_{o_2}(P_{k_v}) & \cdots & \phi_{o_{m-g-1}}(P_{k_v}) & \phi_{o_{m-g}}(P_{k_v}) \end{bmatrix}$$

and

$$Y = \begin{bmatrix} e_{i_1} & 0 & \cdots & 0 \\ 0 & e_{i_2} & \cdots & 0 \\ | & | & & | \\ 0 & 0 & \cdots & e_{i_v} \end{bmatrix}.$$

Suppose the value of $S_{i,j}$ in S are all known, we have the following theorem.

Theorem 1: If column j of S is a partial linear combination of its previous columns with the top $\lfloor (m+1)/2 \rfloor$ components, and if the coefficients are a_i for $1 \leqslant i \leqslant j-1$, then column j is linearly dependent on its previous columns and the error rational points are the roots of

$$f_j - \sum_{i=1}^{j-1} a_i \cdot f_i = 0. \quad (3.3)$$

Proof: See [12, Theorem 1]. □

Unfortunately, the values of $s_{m+1}, \cdots, s_{m+g}$ are unknown, that is, the values of $S_{i,j}$ for $o_{i-1} + o_{j-1} = m+1, m+2, \cdots, m+g$ are unknown, (3.3) may not be found from the matrix S. Thus, the key problem in decoding algebraic–geometric codes is *finding of the real values of* s_i for $i = m+1, m+2, \cdots, m+g$ and (3.3). This problem can be solved by the MFIA developed in Section II. In order to find the values of s_i for $i = m+1, m+2, \cdots, m+g$, our decoding procedure is to find them iteratively, i.e., first find s_{m+1}, then s_{m+2}, and so on, till s_{m+g} is found. In the following, we describe how to find s_{m+w} if we know s_i for $i = 0, o_1, \cdots, m, \cdots, m+w-1$, where $1 \leqslant w \leqslant g$.

Suppose $s_{m+1}, \cdots, s_{m+w-1}$ are found and $s_m + w$ is still unknown, where $1 \leqslant w \leqslant g$. Now we want to find s_{m+w}. Let S' be the rewritten S, where s_p is substituted for $S_{i,j}(o_{i-1} + o_{j-1} = p)$ for $p \leqslant m+w-1$, @ is substituted for $S_{u,v}(o_{u-1} + o_{v-1} = m+w)$, i.e., for s_{m+w}, and # is substituted for $S_{k,h}(o_{k-1} + o_{h-1} > m+w)$, i.e., for s_q with $q > m+w$. Obviously, S' satisfies the requirements of S' in Section II. We will use the MFIA to find the value of @, i.e., the value of s_{m+w}.

Before describing how to find the value of s_{m+w}, we first calculate the number of @, i.e., the number of s_{m+w} in S'. The next lemma is useful.

Lemma 3: Let A_w be the number of nongaps in $[1, w-1]$ for $1 \leqslant w \leqslant g$, then

$$A_w \leqslant \lfloor (w-1)/2 \rfloor. \quad (3.4)$$

Proof: Obviously, it is true for $w = 1$. For $2 \leqslant w \leqslant g$, we consider the two cases

Case 1) If w is a gap, then we consider the two subcases.

- a) w is odd. If $s \in [1, w-1]$, then $w - s \in [1, w-1]$ and $s \neq w - s$, s and $w - s$ are not both nongaps. Thus, (3.4) is true.
- b) w is even. $w/2 \in [1, w-1]$ can not be a nongap. If $s \in [1, w-1]$ and $s \neq w/2$ then $w - s \in [1, w-1]$, $s \neq w - s$, s and $w - s$ are not both nongaps. Thus, $A_w \leqslant (w-2)/2$ and (3.4) is true.

Case 2) If w is a nongap, then from Lemma 2 and $2 \leqslant w \leqslant g$, we can assume that $w, w+1, \cdots, w+p-1$ are nongaps and $w+p$ is a gap, where $1 \leqslant p < g$. From

the previous proof, $A_{w+p} \leqslant \lfloor (w+p-1)/2 \rfloor$. On the other hand, $A_{w+p} = A_w + p$, we have $A_w + p \leqslant \lfloor (w+p-1)/2 \rfloor$ and (3.4) is true. □

Thus, for the number of @, we have the following theorem.

Theorem 2: The number of @ in S^* is at least $m-2g$, that is, at least d^*-2.

Proof: For each $1 \leqslant j \leqslant m-g+1$, from (3.1) there is an @ in column j, if and only if

$$m+w-o_{j-1} \in \{o_0, o_1, \cdots, o_{m-g}\}. \tag{3.5}$$

Equivalently, there is no @ in column j, if and only if

$$m+w-o_{j-1} \notin \{o_0, o_1, \cdots, o_{m-g}\}. \tag{3.6}$$

Let us calculate the number of j, which satisfy (3.6). We consider the following two cases

a) If $m+w-o_{j-1} > m$, i.e., $o_{j-1} < w$, then (3.6) is satisfied. From Lemma 3 the number of such j is $1+A_w$ ($j=1$ satisfies this condition).
b) If $m+w-o_{j-1} \leqslant m$, i.e., $o_{j-1} \geqslant w$, then the number of $m+w-o_{j-1}$ being gaps is B_w, which is the number of the gaps in $[w, 2g-1]$.

From this discussion, the number of columns, in which there are no @, is $1+A_w+B_w$. Since the number of nongaps in $[1, w-1]$ is A_w, the number of gaps in $[1, w-1]$ is $w-1-A_w$. From Lemma 2, we have

$$w-1-A_w+B_w = g. \tag{3.7}$$

Thus, from Lemma 3, we have

$$1+A_w+B_w = 1+A_w+g-w+1+A_w \leqslant 1+g,$$

namely, the number of @ is at least $(m-g+1)-(g+1) = m-2g$. □

Now we are going to explain how to use the MFIA to find the value of s_{m+w}. Some useful facts are introduced as follows.

Lemma 4: If (A) there is a unique value of this @ such that column j of S^* is a partial linear combination of its previous columns with the top i components and (B) column j of S is linearly dependent on its previous columns, then the unique value must be equal to s_{m+w}.

Conversely, from Lemma 1 and Lemma 4, we have the following.

Lemma 5: If (A) is true and (C) the unique value is not equal to s_{m+w}, then column j of S is linearly independent from the previous columns, and when the FIA is applied to S, there is an "×" at (i, j).

For convenience, if (A) is true, then the unique value is called as a *candidate value for* s_{m+w}, or simply, *candidate*. Thus, in order to determine s_{m+w}, our first objective is to find all candidates. In Section II, we have developrd the MFIA to find all candidates. If there is an "×" at (i, j), we say this "×" affects @ in row i and @ in column j. When the MFIA is applied to S^*, we can see the following properties.

a) If there is an "×" in column j, then column j of S is linearly independent on its previous columns.
b) If an @ is at (i, j), then it can be uniquely determined by a partial linear combination of its previous columns with the top i components, if and only if no "×" affects it.
c) An "×" at $(1, 1)$ does not affect any @. An "×" in the first row and not in the first column affects at most one @. An "×" in not in the first column affects at most one @. An "×" neither in the first row nor in the first column affects at most two @.

Now we have the following theorem.

Theorem 3: Applying the MFIA to matrix S^*, there is at least one @, which can be uniquely determined by a partial linear combination, namely, there is at least one candidate value for s_{m+w} calculated by Step 3b).

Proof: Let $t = \lfloor (d^*-1)/2 \rfloor$. Since $v \leqslant t$, there are at most v linearly independent columns in X as well as in S. Suppose there are μ "×" in the discrepancy matrix D after applying the MFIA to S^*. From Property a), $\mu \leqslant v \leqslant t$. We consider the following three cases.

1) If an "×" is at $(1, 1)$, then it does not affect any @, and the other $\mu-1$ "×" affect at most $2(\mu-1)$ @ from Property c). Thus, at most $2\mu-2$ @ are affected.
2) If an "×" in the first column is not in the first row, then it affects at most one @ from Property c). But, there must be another "×" in the first row; this "×" affects at most one @, too (if the received vector has v errors for $1 \leqslant v \leqslant t$, then the known syndromes are not all zero and there must be one "×" in the first row and in the first column, respectively). Thus, the other $\mu-2$ "×" can affect at most $2(\mu-2)$. Hence, at most total $2\mu-2$ @ are affected.
3) If $\mu = 1$, from the discussion in 2), this "×" must be at $(1, 1)$. Thus, there is no @ affected from Property c), that is, $2\mu-2$ (in this case, $2\mu-2=0$) @ are affected.

From Theorem 2, the number of @ is at least d^*-2, at least $d^*-2-(2\mu-2) \geqslant 1$ @ is not affected and it is a candidate from Property b). □

Generally speaking, after applying the MFIA to S^* there are often more than one candidate and they may be not all correct. Fortunately, s_{m+w} can be easily determined from all candidates by the following theorem.

Theorem 4: After applying the MFIA to S^*, the number of correct candidates is greater than the number of the incorrect candidates.

Proof: Suppose that there are μ "×" in the discrepancy matrix D after applying the MFIA to S^*. From the proof of Theorem 3, we can obtain at least $d^*-2-(2\mu-2)$ candidates. If a candidate is equal to s_{m+w}, it is called a correct candidate, otherwise it is called an incorrect candidate. Now we are going to calculate the number of incorrect candidates.

From Property b), the columns of S, which correspond to these columns of S^* containing "×", are linearly independent on their previous columns. The number of these columns is μ. Since the total number of linearly independent columns of S is v or less (because the rank of X is v or less), and from Lemma 5, the number of the incorrect candidates is at most

$v-\mu$. Therefore, at least $d^*-2-2(\mu-1)-(v-\mu)=d^*-v-\mu$ candidates are correct, and the number of the correct candidates is greater than the number of the incorrect candidates ($d^*-v-\mu>v-\mu$). The proof is completed. □

Once s_{m+w} is found, S^* is modified: substituting the correct value of s_{m+w} for @ and substituting @ for s_{m+w+1}. Since the number of s_{m+w+1}, that is, of new @, is d^*-2 or more. If (3.3) is not found, then by the same reason and in the same way, s_{m+w+1} can be found.

However, it is easily seen that to apply the MFIA to the new S^* is equivalent to modifying the discrepancy matrix D as follows: substitute the value of s_{m+w} for all @ being not candidates and eliminate the discrepancy at this place if possible, that is, there is no "×" on its column and there is one "×" on its row; substitute 0 for @ being a correct candidate and "×" for @ being a incorrect candidate (by Lemma 4 and Lemma 5); and place @ at (i,j), at which s_{m+w+1} was. The complexity of modifying discrepancy matrix D is $O((m-g+1)^2)$. Thus, in our decoding procedure the construction of S^* has complexity $O((m-g+1)^2 n)$. The complexity of the algorithm for determining value of s_{m+1} is $O((m-g+1)^3)$. To find $s_{m+2}, s_{m+3}, \cdots, s_{m+g}$ at most $g-1$ modifications of D are required (for some error patterns, (3.3) can be early found), which has complexity $O((g-1)(m-g+1)^2)$. Therefore, to find all s_i for $i=m+1,\cdots,m+g$, and (3.3), the complexity is $O((m-g+1)^2 n)$.

Our decoding procedure can be outlined as follows.

1) Calculate the syndromes s_i from the received vector for $i=o_0,o_1,\cdots,o_{m-g+1}$.
2) Determine s_i for $i=m+1,\cdots,m+g$ and (3.3).
3) Find the roots of (3.3) among the rational points.
4) Solve a system of linear equations and obtain the error locations and error magnitudes at the same time.

Obviously, this decoding procedure is a generalization of Peterson's decoding procedure for the BCH codes. It is easily seen that the complexity of this decoding procedure is $O(n^3)$ where $m \sim n$.

IV. An Example

In this section through an example, we show this simple decoding procedure. Let us consider a plane curve defined by the equation $f(X,Y,Z)=X^5+Y^4Z+YZ^4=0$ over $GF(2^4)$. This curve is called the *Hermitean curve*. With $Q=(0:1:0)$, $x=X/Z$, has pole order 4 and $y=Y/Z$ has pole order 5. From [15] we know that Hermitean curve has the genus $g=6$ and 65 rational points in $GF(2^4)$. Define the code with length 64, i.e., D is all finite rational points, and $G=23Q$. The linear code $C_\Omega(D,G)$ of length 64 over $GF(2^4)$ is the image of the linear map $\alpha^*:\Omega(G-D)\to GF(2^4)^{64}$ defined by

$$\alpha^*(\omega):=(\mathrm{Res}_{P_1}(\omega),\mathrm{Res}_{P_2}(\omega),\cdots,\mathrm{Res}_{P_{64}}(\omega)).$$

From Riemann–Roch theorem, the designed minimum distance $d^*=\deg(G)-2g+2=13$. Any six or fewer errors can be corrected. Since $\deg(G)-g+1=18$, let $f_1,f_2,\cdots,f_{18}$ be a basis of $L(G)$. Then,

$$H=\begin{bmatrix} f_1(P_1) & f_1(P_2) & \cdots & f_1(P_{64}) \\ \vdots & \vdots & \cdots & \vdots \\ f_{18}(P_1) & f_{18}(P_2) & \cdots & f_{18}(P_{64}) \end{bmatrix} \tag{4.1}$$

is a parity check matrix for $C_\Omega(D,G)$.

A basis of $L(G)$ is

$$\begin{array}{lll} f_1=1;(0) & f_2=x;(4) & f_3=y;(5) \\ f_4=x^2;(8) & f_5=xy;(9) & f_6=y^2;(10) \\ f_7=x^3;(12) & f_8=x^2y;(13) & f_9=xy^2;(14) \\ f_{10}=y^3;(15) & f_{11}=x^4;(16) & f_{12}=x^3y;(17) \\ f_{13}=x^2y^2;(18) & f_{14}=xy^3;(19) & f_{15}=y^4;(20) \\ f_{16}=x^4y;(21) & f_{17}=x^3y^2;(22) & f_{18}=x^2y^3;(23) \end{array} \tag{4.2}$$

where the number in parenthesis indicates the pole order of the corresponding function with Q.

For convenience, let $\phi_0=f_1$, $\phi_4=f_2$, $\phi_5=f_3$, $\phi_8=f_4$, $\phi_9=f_5$, $\phi_{10}=f_6$, and $\phi_i=f_{i-5}$ for $i=12,13,\cdots,23$. Thus,

$$\mathrm{ord}_Q(\phi_i)=-i, \qquad \text{for } i=0,4,5,8,9,10,12,13,\cdots,23.$$

For each i, there may be many ϕ such that $\mathrm{ord}_Q(\phi_i)=-i$. But it is sure that these ϕ must be linear combinations of $\phi_0,\phi_3,\phi_5,\cdots,\phi_i$. For example, $\mathrm{ord}_Q(x^5)=-20$. Since $X^5+Y^4Z+YZ^4=0$, we have

$$x^5=y^4+y,$$
$$x^6=xy^4+xy,$$

and so on.

Let $\phi_{20}=y^4$ and $\phi'_{20}=x^5$, $\phi_{24}=xy^4$, $\phi'_{24}=x^6$, and so on. Then, from the previous equations, we have

$$\phi'_{20}=\phi_{20}+\phi_5, \qquad \phi'_{24}=\phi_{24}+\phi_9. \tag{4.3}$$

Let $u=(u_1,u_2,\cdots,u_{64})$ be a received word, $e=(e_1,e_2,\cdots,e_{64})$ be an error vector, and $c=(c_1,c_2,\cdots,c_{64})$ be a codeword. Thus, we have $u=e+c$. Then, we define the syndromes

$$\begin{aligned} s_i &:= \langle u,\alpha^*\phi_i\rangle \\ &= \sum_{j=1}^{64} u_j\phi_i(P_j), \qquad \text{for } i=0,4,5,8,9,10,12,13,\cdots,23; \\ &= \sum_{j=1}^{64} e_j\phi_i(P_j), \end{aligned} \tag{4.4}$$

$$\begin{aligned} s'_i &:= \langle u,\alpha^*\phi'_i\rangle = \sum_{j=1}^{64} u_j\phi'_i(P_j), \qquad \text{for } i=20 \\ &= \sum_{j=1}^{64} e_j\phi'_i(P_j). \end{aligned} \tag{4.5}$$

Let

$$s_i=\sum_{j=1}^{64} e_j\phi_i(P_j) \text{ and } s'_i=\sum_{j=1}^{64} e_j\phi'_i(P_j) \qquad \text{for } i\geq 24,$$

where ϕ_i and ϕ_i' have order $-i$ with Q. Certainly, s_i and s_i' for $i \geq 24$ are unknown. From (4.3)–(4.5), we know that

$$s_{20}' = s_{20} + s_5, \qquad s_{24}' = s_{24} + s_9. \tag{4.1}$$

Suppose there are v errors e_{k_μ} at $P_{k_\mu} = (X_{k_\mu} : Y_{k_\mu} : Z_{k_\mu})$ for $\mu = 1, 2, \cdots, v$, where $v \leqslant 6$.

Consider a received vector

$$u = (\alpha^{12}, \alpha^4, \alpha^7, \alpha^8, \alpha^9, \alpha^9, 0, \cdots, 0, 0, 0, 0, \cdots, 0, \\ 0, 0, 0, 0),$$

that is, $u_1 = \alpha^{12}$, $u_2 = \alpha^4$, $u_3 = \alpha^7$, $u_4 = \alpha^8$, $u_5 = \alpha^9$, $u_6 = \alpha^9$, and other components are 0. Let $P_1 = (1 : 1 : \alpha)$, $P_2 = (1 : 1 : \alpha^2)$, $P_3 = (1 : 1 : \alpha^4)$, $P_4 = (1 : 1 : \alpha^8)$, $P_5 = (0 : 0 : 1)$, $P_6 = (1 : \alpha : 1)$.

From (4.2) and (4.4), we have

$$\begin{array}{llll} s_0 = \alpha & s_4 = \alpha^{14} & s_5 = \alpha^2 & s_6 = \alpha^{11} \\ s_9 = \alpha^4 & s_{10} = \alpha^9 & s_{12} = \alpha^4 & s_{13} = \alpha^{11} \\ s_{14} = \alpha^{10} & s_{15} = \alpha^5 & s_{16} = 1 & s_{17} = \alpha^6 \\ s_{18} = \alpha^8 & s_{19} = \alpha^2 & s_{20} = \alpha^5 & s_{21} = \alpha^{12} \\ s_{22} = \alpha^5 & s_{23} = \alpha^{10} & s_{20}' = \alpha. & \end{array}$$

In order to find s_{24} (and s_{24}'), we construct S^* as shown in the first matrix at bottom of page where @ expresses s_{24} or s_{24}' and # expresses other unknown syndromes. In this example, S^* is shown in the second matrix at bottom of page.

Applying the MFIA to S^*, we have the following discrepancy matrix, where the candidate in column j is denoted by $@_j$

$$\begin{bmatrix} s_0 & s_4 & s_5 & s_8 & s_9 & s_{10} & s_{12} & s_{13} & s_{14} & s_{15} & s_{16} & s_{17} & s_{18} & s_{19} & s_{20} & s_{21} & s_{22} & s_{23} \\ s_1 & s_5 & s_9 & s_{12} & s_{13} & s_{14} & s_{16} & s_{17} & s_{18} & s_{19} & s_{20} & s_{21} & s_{22} & s_{23} & @_1 & \# & \# & \# \\ s_5 & s_9 & s_{10} & s_{13} & s_{14} & s_{15} & s_{17} & s_{18} & s_{19} & s_{20}' & s_{21} & s_{22} & s_{23} & @_1 & \# & \# & \# \\ s_8 & s_{12} & s_{13} & s_{16} & s_{17} & s_{18} & s_{20}' & s_{21} & s_{22} & s_{23} & @_1 & \# & \# & \# & \# & \# & \# & \# \\ s_9 & s_{13} & s_{14} & s_{17} & s_{18} & s_{19} & s_{21} & s_{22} & s_{23} & @_1 & \# & \# & \# & \# & \# & \# & \# & \# \\ s_{10} & s_{14} & s_{15} & s_{18} & s_{19} & s_{20} & s_{22} & s_{23} & @_1 & \# & \# & \# & \# & \# & \# & \# & \# & \# \\ s_{12} & s_{16} & s_{17} & s_{20}' & s_{21} & s_{22} & @_1 & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\ s_{13} & s_{17} & s_{18} & s_{21} & s_{22} & s_{23} & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\ s_{14} & s_{18} & s_{19} & s_{22} & s_{23} & @_1 & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\ s_{15} & s_{19} & s_{20} & s_{23} & @_1 & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\ s_{16} & s_{20}' & s_{21} & @_1 & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\ s_{17} & s_{21} & s_{22} & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\ s_{18} & s_{22} & s_{23} & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\ s_{19} & s_{23} & @_1 & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\ s_{20} & @_1 & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\ s_{21} & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\ s_{22} & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\ s_{23} & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \end{bmatrix}$$

$$\begin{bmatrix} \alpha & \alpha^{14} & \alpha^2 & \alpha^{11} & \alpha^4 & \alpha^9 & \alpha^4 & \alpha^{11} & \alpha^{10} & \alpha^5 & 1 & \alpha^6 & \alpha^8 & \alpha^2 & \alpha^5 & \alpha^{12} & \alpha^5 & \alpha^{10} \\ \alpha^{14} & \alpha^{11} & \alpha^4 & \alpha^4 & \alpha^{11} & \alpha^{10} & 1 & \alpha^6 & \alpha^8 & \alpha^2 & \alpha & \alpha^{12} & \alpha^5 & \alpha^{10} & @_1 & \# & \# & \# \\ \alpha^2 & \alpha^4 & \alpha^9 & \alpha^{11} & \alpha^{10} & \alpha^5 & \alpha^6 & \alpha^8 & \alpha^2 & \alpha^5 & \alpha^{12} & \alpha^5 & \alpha^{10} & @_1 & \# & \# & \# & \# \\ \alpha^{11} & \alpha^4 & \alpha^{11} & 1 & \alpha^6 & \alpha^8 & \alpha & \alpha^{12} & \alpha^5 & \alpha^{10} & @_1 & \# & \# & \# & \# & \# & \# & \# \\ \alpha^4 & \alpha^{11} & \alpha^{10} & \alpha^6 & \alpha^8 & \alpha^2 & \alpha^{12} & \alpha^5 & \alpha^{10} & @_1 & \# & \# & \# & \# & \# & \# & \# & \# \\ \alpha^9 & \alpha^{10} & \alpha^5 & \alpha^8 & \alpha^2 & \alpha^5 & \alpha^8 & \alpha^{10} & @_1 & \# & \# & \# & \# & \# & \# & \# & \# & \# \\ \alpha^4 & 1 & \alpha^6 & \alpha & \alpha^{12} & \alpha^5 & @_1 & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\ \alpha^{11} & \alpha^6 & \alpha^8 & \alpha^{12} & \alpha^5 & \alpha^{10} & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\ \alpha^{10} & \alpha^8 & \alpha^2 & \alpha^5 & \alpha^{10} & @_1 & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\ \alpha^5 & \alpha^2 & \alpha^5 & \alpha^{10} & @_1 & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\ 1 & \alpha & \alpha^{12} & @_1 & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\ \alpha^6 & \alpha^{12} & \alpha^5 & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\ \alpha^8 & \alpha^5 & \alpha^{10} & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\ \alpha^2 & \alpha^{10} & @_1 & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\ \alpha^5 & @_1 & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\ \alpha^{12} & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\ \alpha^5 & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\ \alpha^{10} & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \end{bmatrix}$$

and "×" in column j is denoted by $\times_j$, and their values are in the first matrix at bottom of page where $\times_1 = \alpha$, $\times_2 = 1$, and $\times_3 = \alpha^3$; $@_4 = @_7 = @_{11} = \alpha^{14}$, $@_5 = @_6 = @_9 = @_{10} = \alpha^8$. From Theorem 4, $s_{24} = s_{31} = s_{26} = s_{29} = s_{210} = \alpha^8$ are correct and $s_{04} = s_{27} = s_{211} = \alpha^{14}$ are not correct. s'_{24} should be α^9.

Now we modify the discrepancy matrix as follows. Substitute α^8 (or α^9) for all @ being not candidates and eliminate the discrepancies at these places if it is possible (the complexity is at most $O(t^2)$, then substitute @ for s_{25} (or s'_{25}). Then, place 0 at (5,10), (6,9), (9,6), and (10,5), respectively (the candidate values for s_{24} at these locations were correct) and place $\times_4$ at (11,4), $\times_7$ at (7,7), and $\times_{11}$ at (4, 11), (the candidate values for s'_{24} are not correct). The new matrix is equivalent to that the MFIA has been applied to new S^*, where the values of s_{24} and s'_{24} are known and the values of s_{23} and s'_{25} are unknown. So to find s_{25} and s'_{25}, that is, calculate the values of candidates of all s_{25} and eliminates of discrepancies at some places, where s_{24} were. After this discrepancy matrix is modified, we have the second matrix at bottom of page where $\times_1 = \alpha$, $\times_2 = 1$, $\times_3 = \alpha^3$, $\times_4 = \times_7 = \times_{11} = \alpha^{12}$; $s_{25} = @_6 = @_9 = 1$.

By the same way we can find that $s_{26} = \alpha$, $s_{27} = \alpha^4$, $s_{28} = 1$, and $s_{29} = \alpha^8$. Thus, we can find two solutions of (3.3). They are linear dependent relations of column 5 with the first 4 columns and column 6 with the first 4 columns, respectively. We have

$$xy + x^2 + y + x = 0.$$
$$y^2 + x^2 + \alpha^4 x + \alpha^4 y = 0.$$

$$\begin{bmatrix}
\times_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\alpha^{11} & \times_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & @ & \# & \# \\
\alpha^2 & \alpha^4 & \times_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & @ & \# & \# & \# \\
\alpha^{11} & \alpha^4 & \alpha^{11} & 0 & 0 & 0 & 0 & 0 & 0 & @_{11} & \# & \# & \# & \# & \# & \# \\
\alpha^4 & \alpha^{11} & \alpha^{10} & 0 & 0 & 0 & 0 & 0 & @_{10} & \# & \# & \# & \# & \# & \# & \# \\
\alpha^9 & \alpha^{10} & \alpha^5 & 0 & 0 & 0 & 0 & @_9 & \# & \# & \# & \# & \# & \# & \# & \# \\
\alpha^4 & 1 & \alpha^6 & 0 & 0 & 0 & @_8 & \# & \# & \# & \# & \# & \# & \# & \# & \# \\
\alpha^{11} & \alpha^6 & \alpha^8 & 0 & 0 & 0 & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\
\alpha^{10} & \alpha^8 & \alpha^2 & 0 & 0 & @_6 & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\
\alpha^5 & \alpha^2 & \alpha^5 & 0 & @_5 & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\
1 & \alpha & \alpha^{12} & @_4 & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\
\alpha^6 & \alpha^{12} & \alpha^5 & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\
\alpha^8 & \alpha^5 & \alpha^{10} & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\
\alpha^2 & \alpha^{10} & @ & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\
\alpha^5 & @ & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\
\alpha^{12} & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\
\alpha^5 & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\
\alpha^{10} & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \#
\end{bmatrix}$$

$$\begin{bmatrix}
\times_1 & 0 & 0 & 0 & 0 & 0 & 0. & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\alpha^{14} & \times_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & @ & \# & \# \\
\alpha^2 & \alpha^4 & \times_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & @ & \# & \# & \# \\
\alpha^{11} & \alpha^4 & \alpha^{11} & 0 & 0 & 0 & 0 & 0 & 0 & \times_{11} & @ & \# & \# & \# & \# & \# & \# \\
\alpha^4 & \alpha^{11} & \alpha^{10} & 0 & 0 & 0 & 0 & 0 & 0 & @ & \# & \# & \# & \# & \# & \# & \# \\
\alpha^9 & \alpha^{10} & \alpha^5 & 0 & 0 & 0 & 0 & 0 & @_{10} & \# & \# & \# & \# & \# & \# & \# & \# \\
\alpha^4 & 1 & \alpha^6 & 0 & 0 & 0 & \times_7 & @ & \# & \# & \# & \# & \# & \# & \# & \# & \# \\
\alpha^{11} & \alpha^6 & \alpha^8 & 0 & 0 & 0 & @ & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\
\alpha^{10} & \alpha^8 & \alpha^2 & 0 & 0 & 0 & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\
\alpha^5 & \alpha^2 & \alpha^5 & 0 & 0 & @_6 & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\
1 & \alpha & \alpha^{12} & \times_4 & @ & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\
\alpha^6 & \alpha^{12} & \alpha^5 & @ & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\
\alpha^8 & \alpha^5 & \alpha^{10} & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\
\alpha^2 & \alpha^{10} & \alpha^8 & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\
\alpha^5 & \alpha^8 & @ & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\
\alpha^{12} & @ & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\
\alpha^5 & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\
\alpha^{10} & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \# & \#
\end{bmatrix}$$

known. So
ndidates of
ces, where
l, we have
$x, x_3 = 1$,
$_0 = 1$.
$x_{27} = \alpha^4$,
olutions of
mn 5 with
i columns,

The two equations have six common roots, the five points on the line $y = x$:

$$P_1 = (1:1:\alpha), \quad P_2 = (1:1:\alpha^2),$$
$$P_3 = (1:1:\alpha^4), \quad P_4 = (1:1:\alpha^8),$$
$$P_5 = (0:0:1),$$

and the point of intersection of the two lines $x + 1 = 0$ and $y + x + \alpha^4$:

$$P_6 = (1:\alpha:1).$$

In order to find the real error locations and the error magnitudes, we should solve reduced parity check equations in the six unknowns e_1, e_2, e_3, e_4, e_5, and e_6. Solving these linear equations, we have

$$e_1 = \alpha^{12}; \quad e_2 = \alpha^4;$$
$$e_3 = \alpha^7; \quad e_4 = \alpha^8;$$
$$e_5 = \alpha^9; \quad e_6 = \alpha^9.$$

Thus, there are six errors. They are at P_1, P_2, P_3, P_4, P_5, and P_6 and the error magnitudes are α^{12}, α^4, α^7, α^8, α^9, and α^9, respectively.

V. Conclusion

In this paper, we have derived a very simple decoding procedure for decoding algebraic–geometric codes $C_\Omega(D, G)$ with $G = m \cdot Q$ up to $\lfloor (d^* - 1)/2 \rfloor$ errors. The computation complexity of this decoding procedure is $O(n^3)$. This decoding procedure is a generalization of Peterson's decoding procedure. It should be noted that the decoding procedure is applicable to decoding some cyclic codes up to the van Lint–Wilson's bound and decoding some cyclic codes beyond actual minimum distance.

Acknowledgment

The authors are deeply grateful to J. R. Myrick, Jr. and J. Kralik for helpful discussion and comments concerning this work. The authors would like to thank the referees, the Associate Editor, Dr. A. Tietavainen, and I. M. Duursma for their many valuable suggestions on the style and presentation of this paper. Mr. Duursma found some minor typing errors in the early version of this paper and suggested the example in this new version. He also obtained a different proof of our results during the review process and generalized the decoding procedure to the case of arbitrary G during revision of the paper.

References

[1] M. A. Tsfasman, S. G. Vlăduţ, and T. Zink, "Modular curves, Shimura curves and Goppa codes, better than Varshamove–Gilbert bound," *Math. Nachr.*, vol. 104, pp. 13–28, 1982.
[2] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: North Holland, 1977.
[3] G. L. Katsman, M. A. Tsfasman, and S. G. Vlăduţ, "Modular curves and codes with a polynomial construction," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 353–355, Mar. 1984.
[4] J. Wolfmann, "Recent results on coding and algebraic geometry," in *Proc. 3rd Int. Conf. AAECC-3*, Grenobe, France, July 1985, pp. 167–184.
[5] Y. Driencourt, "Some properties of elliptic codes over a field of characteristic 2," in *Proc. 3rd Int. Conf. AAECC-3*, Grenobe, France, July 1985, pp. 185–193.
[6] Y. Driencourt and J.F. Michon, "Elliptic codes over field of characteristic 2," *J. Pure and Appl. Algebra*, vol. 45, pp. 15–39, Mar. 1987.
[7] H. J. Tiersma, "Remarks on codes from Hermitian curves," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 605–609, Sept. 1987.
[8] J. H. van Lint and T. A. Springer, "Generalized Reed–Solomon codes from algebraic geometry," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 305–310, May 1987.
[9] J. P. Hensen, "Codes on the Klein quartic, ideals and decoding," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 919–923, Nov. 1987.
[10] S. Harari, "New codes from algebraic curves of genus 2," presented at the *IEEE Int. Symp. Inform. Theory*, Ann Arbor, MI, Oct. 1986.
[11] J. Justesen, K. J. Larsen, H. Elbrond Jensen, and T. Høholdt, "Construction and decoding of a class of algebraic geometry codes," *IEEE Trans. Inform. Theory*, vol. 35, pp. 811–821, July 1989.
[12] A. N. Skorobogatov and S. G. Vlăduţ, "On decoding of algebraic–geometric codes," *IEEE Trans. Inform. Theory*, vol. 36, pp. 1051–1060, Sept. 1990.
[13] R. Pellikaan, "On a decoding algorithm for codes on maximal curves," *IEEE Trans. Inform. Theory*, vol. 35, pp. 1228–1232, Nov. 1989.
[14] J. Justesen, K. J. Larsen, H. Elbrond Jensen, and T. Høholdt, "Fast decoding of codes from algebraic plane curves," *IEEE Trans. Inform. Theory*, vol. 38, pp. 111–119, Jan. 1992.
[15] J. H. van Lint, "Algebraic geometry codes," in *Coding Theory and Design Theory*, vol. 20 (IMA Volumes in Mathematics and its Applications). New York: Springer, 1988, pp. 137–162.
[16] G. L. Feng and K. K. Tzeng, "A generalization of the Berlekamp–Massey algorithm for multisequence shift-register synthesis with applications to decoding cyclic codes," *IEEE Trans. Inform. Theory*, vol. 37, pp. 1274–1287, Sept. 1991.